The Problem of Normal Form for Unlabeled Boundary NLC Graph Languages

数理解析研究所講究録
第 796 巻 1992 年 84-92

The Problem of Normal Form for Unlabeled Boundary NLC Graph Languages

Koichi Yamazaki *　　Takeo Yaku †

Abstract

In the previous paper (Rozenberg and Welzl 1986 a), it is open that whether there exists a positive integer k_0 such that there is a BNLC grammar G with $maxr(G) \leq k_0$ and $L = und(L(G))$ for every unlabeled BNLC language L, where $maxr(G)$ is the maximum number of nodes productions in G, and $und(L(G))$ is the set of underlying unlabeled graphs which are obtained from graphs in $L(G)$ by taking off the labels. In order to negatively solve this open problem, we first show a pumping lemma for a BNLC languages. Then we will show that there is no integer k_0 satisfying the above conditions, using the pumping lemma.

1 Introduction

NLC graph grammars are introduced by Janssens and Rozenberg as a basic framework for the mathematical investigation of graph grammars. Since then this model has been intensively investigated (see, e.g., Janssens and Rozenberg 1980, Ehrenfeucht et al 1984 and Janssens et al 1984). BNLC graph grammars are introduced and investigated by Rozenberg and Welzl in (Rozenberg and Welzl 1986 a,b). BNLC graph grammars are NLC graph grammars with the property whenever -in a graph already generated- two nodes may be rewritten, then those nodes are not adjacent. BNLC languages are an attractive subfamily of the family of NLC languages (see 1986 b).

As is necessary in string grammars, in graph grammars it is important to investigate normal forms for grammars. Let G be a context-free string grammar, and let $maxr_{st}(G)$ be the maximum length of the right side of the productions in G. "Whether there is a positive integer k_0, such that for every context-free string language there is a context-free string grammar G with $maxr_{st}(G) \leq k_0$ and $L = L(G)$

*NEC Corporation
†Tokyo Denki University

?". The answer of the above problem for $k_0 = 2$ is well known as the Chomsky normal form. In this paper, we call the above problem the Chomsky normal form for context-free string grammars. By Ehrenfeucht et al (1984), for BNLC graph grammar the following problem was investigated : "Whether there is a positive integer k_0, such that for every NLC graph language L there is a NLC graph grammar G with $maxr(G) \leq k_0$ and $L = L(G)$?", where $maxr(G)$ is maximum of the number of nodes of axiom and graphs of right hand side of productions in G. In (Ehrenfeucht et al 1984), it was shown that there is no positive integer k_0 such that k_0 satisfy the above condition. This problem is similar to the Chomsky normal form for context-free string grammars. In this paper, we call the above problem the Chomsky normal form problem for NLC graph grammars.

In (Rozenberg and Welzl 1986 a), the Chomsky normal form problem for BNLC graph grammar was investigated : "Whether there is a positive integer k_0 such that for every BNLC graph language L there is a BNLC graph grammar G with $maxr(G) \leq k_0$ and $L = L(G)$?". In (Rozenberg and Welzl 1986 a), it was shown that for every BNLC graph grammar there is no positive integer k_0 such that k_0 satisfies the above condition as was previously shown in the NLC case. In (Rozenberg and Welzl 1986 a), however it is an open problem that whether there is the Chomsky normal form for unlabeled BNLC languages, i.e., " Whether there is a positive integer k_0, such that for every unlabeled BNLC language L there is a BNLC grammar G with $maxr(G) \leq k_0$ and $L = und(L(G))$?", where $und(L(G))$ is the set of underlying unlabeled graphs of $L(G)$. It turns out that there exists no positive integer k_0 of this problem. In this paper, by pumping lemma for NLC languages, we will show that there exists no the Chomsky normal form for unlabeled BNLC languages, i.e., there exists no positive integer k_0 that satisfy the following condition: For every unlabeled BNLC language L, there is a BNLC graph grammar G such that $maxr(G) \leq k_0$ and $L = und(L(G))$.

This paper is organized as follows. In Section 2, definitions basic notions. In Section 3, pumping lemma for BNLC languages are given. In Section 4, for every k, we construct a u-BNLC language L_k such that L_k is never constructed by a BNLC graph grammar with $maxr(G) \leq k$, and give properties of L_k. In Section 5, it is shown, using the pumping lemma and L_k, that there exists no the Chomsky normal form for unlabeled BNLC languages.

2 Preliminaries

We start with basic notations concerning graphs, graph grammars and concrete derivation which we need for this paper. For details, see (1986 a). We based on the definition in (1986 a). We assume familiarity with elementary graph theory. In particular, we use the following notions as defind in Harary (1969): adjacent, neighbor, subgraphs, induced subgraphs.

2.1 Graphs

We consider finite undirected node labeled graphs without loops and without multiple edges. For a set of labels Σ, a graph X (over Σ) is specified by a finite set V_X of nodes, a set E_X of two element subsets of V_X (called the set of edges) and a function φ_X from V_X into Σ (called the *labeling function*). Notice that E_X is a set of sets of the elements of V_X. Disregarding the labeling function one gets the *underlying unlabeled graph of* X, denoted by $und(X)$. The set of all graphs over Σ is denoted by G_Σ. The graph $X-x$ is the subgraph of X induced by $V_X - \{x\}$. The *neighborhood of* x *in* X, $nbh_X(x)$, is the set $\{\varphi_X(y) \mid \{x,y\} \in E_X\}$. A graph X' is *isomorphic* to X, if there is a bijection from $V_{X'}$ to V_X which preserves labels and adjacencies. The set of all graphs isomorphic to X is denoted by $[X]$. The size of X, $\sharp X$ is the number of nodes in X. We also denote the cardinality of V_X by $\sharp V_X$, i.e., $\sharp X = \sharp V_X$.

Let Φ be a set of labels. A graph X is called a Φ - *boundarygraph*, if no two nodes of X that are labeled by elements of Φ are adjacent.

2.2 Graph Grammars

A *node label controlled* (NLC) *grammar* is a system $G = (\Sigma, \Delta, P, conn, Z_{ax})$, where Σ is a finite nonempty set of *labels*, Δ is a nonempty subset of Σ (the set of *terminals*), P is a finite set of pairs (d,Y) where $d \in \Sigma$ and $Y \in G_\Sigma$ (the set of *productions*), *conn* is a function from Σ into 2^Σ (the *connection function*), and $Z_{ax} \in G_\Sigma$ (the *axiom*).

By $[P]$ we denote the set $\{(d,Y') \mid Y' \in [Y] \text{ for some } (d,Y) \in P\}$. By $maxr(G)$ we denote $max(\{\sharp Z_{ax}\} \cup \{\sharp Y \mid (d,Y) \in P \text{ for some } d \in \Sigma\})$. The set $\Sigma - \Delta$ is refered to as the set of *nonterminals*. We define the set of nonterminal nodes by Γ, i.e., $\Gamma = \Sigma - Delta$. In the context of G, given a graph $X \in G_\Sigma$ we refer to nodes labeled by elements of Γ (Δ, respectively) as *nonterminal nodes* (*terminal nodes*, respectively).

Let X, Y and Z be graphs over Σ with $V_X \cap V_Y = \emptyset$ and let $x \in V_X$. Then X *concretely derives* Z (*in* G, *replacing* x *by* Y), denoted by $X \Rightarrow_{G\ (x,Y)} Z$ or simply by $X \Rightarrow_{(x,Y)} Z$,if $(\varphi_X(x), Y) \in [P]$, $V_Z = V_{X-x} \cup V_Y$, $E_Z = E_{X-x} \cup E_Y \cup \{\{x',y\} \mid x' \in nbh_X(x),\ y \in V_Y,\ \varphi_X(x') \in conn(\varphi_Y(y))\}$, φ_Z equals to φ_{X-x} on V_{X-x}, and φ_Z equals to φ_Y on V_Y. Intuitively speaking, we replace x in X by the graph Y and connect a node y of Y to a neighbor x' of x if and only if $\varphi_X(x') \in conn(\varphi_Y(y))$.

A graph X *directly derives* a graph Z (*in* G), in symbols $X \Rightarrow_G Z$, if there is a graph $Z' \in [Z]$, such that X concretely derives Z' in G. $\Rightarrow_G^*$ is the transitive and reflexive closure of $\Rightarrow_G$. If $X \Rightarrow_G^* Z$, then we say that X *derives* Z (*in* G). The *language of* G is the set $\{X \in G_\Delta \mid Z_{ax} \Rightarrow_G^* X\}$. A set L of graphs is an NLC *language* if there is an NLC grammar G such that $L = L(G)$. A *boundary* NLC (*BNLC*) *grammar* is an NLC grammar $G = (\Sigma, \Delta, P, conn, Z_{ax})$, where Z_{ax} is a Γ - boundary graph and for all $(d,Y) \in P$, $d \in \Gamma$ and Y is a Γ - boundary graph. A graph language L is an NLC (BNLC) *language*, if there is a NLC (BNLC) grammar

G such that $L = L(G)$. A language L of unlabeled graphs is *u-NLC* (*u-BNLC*) *language*, if there is an NLC (BNLC) language L' such that $L = und(L')$. Let G is NLC graph grammar. G is *chain-free*, if for all $(d, Y) \in P$ with $V_Y = \{y\}$ (i.e., $\sharp Y = 1$), y is a terminal node.

2.3 Concrete Derivation

Let $G = (\Sigma, \Delta, P, conn, Z_{ax})$ be an NLC grammar. If a graph X concretely derives a graph Z in G, replacing a node x by a graph Y, then we refer to the construct $X \Rightarrow_{(x,Y)} Z$ as a *concrete derivation step in* G (*from* X *to* Z). A sequence of successive concrete derivation steps in G

$D : X_0 \Rightarrow_{(x_0,Y_1)} X_1 \Rightarrow_{(x_1,Y_2)} \cdots \Rightarrow_{(x_{n-1},Y_n)} X_n,$

where $n \geq 0$ and the sets V_{X_0}, V_{Y_i}, $1 \leq i \leq n$, are pairwise disjoint, is refered to as a *concrete derivation in* G (*from* X_0 *to* X_n).

The *node set of* D is $V_D = \bigcup_{i=0}^{n} V_{X_i}$. The *edge set of* D is $E_D = \bigcup_{i=0}^{n} E_{X_i}$. The *labeling function* φ_D *of* D is defined by $\varphi_D(x) = \varphi_{X_0}(x)$ if $x \in V_{X_0}$ and $\varphi_D(x) = \varphi_{Y_i}(x)$ if $x \in V_{Y_i}$, for some i, $1 \leq i \leq n$. Note that $V_D = V_{X_0} \cup \bigcup_{i=1}^{n} V_{Y_i}$, hence φ_D is defined on the whole set V_D. Moreover, if $x \in V_{X_i}$, for some i, $1 \leq i \leq n$, then $\varphi_{X_i}(x) = \varphi_D(x)$. Thus every concrete derivation D defines naturally a graph with set of nodes V_D, set of edges E_D and labeling function φ_D. Note that this graph D is a Γ-boundary graph whenever X_0 is a Γ-boundary graph and G is a BNLC grammar.

Let O_D be a distinguished element not in V_D which is called the *origin of* the derivation D. The *predecessor mapping* $pred_D$ *of* D is a function from V_D into $V_D \cup \{O_D\}$ such that for $x \in V_D$

$$pred_D(x) = \begin{cases} O_D & \text{if } x \in V_{X_0} \text{ and} \\ x_i & \text{if } x \in V_{Y_{i+1}} \text{ for an } i,\ 0 \leq i \leq n-1 \end{cases}$$

Hence $pred_D$ maps every node x in V_D to the node from which x is directly derived (or to O_D if x was already present in X_0).

The *history* $hist_D(x)$ *of a node* $x \in V_D$ in D is the sequence $(y_0, y_1, \cdots, y_m)$, $m \geq 1$, $y_i \in V_D$ for all i, $1 \leq i \leq m$, such that $y_0 = O_D$, $y_m = x$, and $y_i = pred_D(y_{i+1})$ for all i, $0 \leq i \leq m-1$. Let D be a derivation and let x and y be nodes in V_D. A node x is an *ancestor of* y *in the derivation* D if $x \in hist_D(y)$. Nodes x and y are *independent* if $x \notin hist_D(y)$ and $y \notin hist_D(x)$. Let $(y_0, y_1, \cdots, y_m)$ be a sequence such that $hist_D(x) = (y_0, y_1, \cdots, y_m)$, and let $0 \leq i < j \leq m$. Then we denote the sequence $(y_i, y_{i+1}, \cdots, y_j)$ by $hist_D(y_i, y_j)$. (We can define $hist_D(x, y)$ only when x is an ancestor of y). Let

$$D : X_0 \Rightarrow_{(x_0,Y_1)} X_1 \Rightarrow_{(x_1,Y_2)} \cdots \Rightarrow_{(x_{n-1},Y_n)} X_n$$

be a derivation. We denote the set $\{x_0, x_1, \cdots, x_{n-1}\}$ of rewritten nonterminal nodes in derivation D by C_D. We call the graph X_n the result in the derivation D.

Let D' be derivation

$$D' : X'_0 \Rightarrow_{(x'_0,Y'_1)} X'_1 \Rightarrow_{(x'_1,Y'_2)} \cdots \Rightarrow_{(x'_{n-1},Y'_n)} X'_n.$$

The derivation D' is isomorphic to D if there is a bijection h from $V_{D'}$ to V_D such that $h\,|_{V_{X'_i}}$ is an isomorphism from $V_{X'_i}$ to V_{X_i} and for $x' \in V_{X'_i}$ $(0 \le i \le n-1)$, $h(pred_{D'}(x')) = pred_D(h(x'))$. The set of all derivations isomorphic to D is denoted by [D].

Let D be a derivation.

$$D : X_0 \Rightarrow_{(x_0,Y_1)} X_1 \Rightarrow_{(x_1,Y_2)} \cdots \Rightarrow_{(x_{n-1},Y_n)} X_n$$

We call the graph X_n the *result* in the derivation D.

3 A pumping lemma for BNLC languages

In this section, we introduce a pumping lemma for BNLC grammars. In this paper we need the pumping lemma is the proof of the main theorem.

In order to state the pumping lemma, we need to develop some concepts. Let $G = (\Sigma, \Delta, P, conn, Z_{ax})$ be a BNLC grammar. Let D be a following concrete derivation in G :

$$D : X_0 \Rightarrow_{(x_0,Y_1)} X_1 \Rightarrow_{(x_1,Y_2)} \cdots \Rightarrow_{(x_{n-1},Y_n)} X_n,$$

such that there exist $x_p, x_q \in C_D$ $(x_p \neq x_q)$ satisfying $x_p \in hist_D(x_q)$ and $\varphi_D(x_p) = \varphi_D(x_q)$. We call the above derivation *rough derivation on* (x_p, x_q).

Let us consider the following derivation $\mathcal{D}$,

$$\mathcal{D} : X'_0 \Rightarrow_{(x_{i(0)},Y_{i(0)+1})} X'_1 \Rightarrow_{(x_{i(1)},Y_{i(1)+1})} \cdots \Rightarrow_{(x_{i(e-1)},Y_{i(e-1)+1})} X'_e,$$

where the sequence $(i(0), i(1), \cdots, i(e-1))$ is a sequence ascending order of the subscripts in the set $\{x_l \in C_D \mid x_p \in hist_D(x_l) \text{ and } x_q \notin hist_D(x_l)\}$.

In what follows, we will show that the derivation $\mathcal{D}$ can be iterated in this section. Let m be a positive integer. Let $D'_1, D'_2, \cdots, D'_m \in [\mathcal{D}]$ be derivations, and h_j be an isomorophism from D'_j to $\mathcal{D}$ for each j, $1 \le j \le m$ of the following forms:

$$\begin{aligned} D'_j \ : \ X^j_0 \ &\Rightarrow_{(x^j_{i(0)},Y^j_{i(0)+1})} X^j_1 \\ &\Rightarrow_{(x^j_{i(1)},Y^j_{i(1)+1})} X^j_2 \\ &\qquad \vdots \\ &\Rightarrow_{(x^j_{i(e-1)},Y^j_{i(e-1)+1})} X^j_e \end{aligned}$$

$h_j \ : \ V_{D'_j} \to V_{iter(D,x_p,x_q)}$, $x^j_l \mapsto x_l$, where $l \in \{i(0), i(1), \cdots, i(e-1)\}$, which satisfy the following four conditions:

(1) $V_{iter(D,x_p,x_q)} \cap V_{D'_1} = \{x_q\}$ and $x_q = x^1_{i(0)}$,
(2) For each $1 \leq j \leq m-1$, $V_{D'_j} \cap V_{D'_{j+1}} = \{x^j_q\}$ and $x^j_q = x^j_{i(0)}$,
(3) $V_{D'_{m-1}} \cap V_{D'_m} = \{x^{m-1}_q\}$ and $x^{m-1}_q = x^m_{i(0)}$, and
(4) For every i, j such that $| i-j | \geq 2$, $V_{D'_i} \cap V_{D'_j} = \emptyset$.

Then we obtain the following derivation :

$$
\begin{aligned}
X''_0 \quad & \Rightarrow_{(x_0,Y_1)} X''_1 \cdots \Rightarrow_{(x_{q-1},Y_q)} X''_q \\
& \qquad \vdots \\
& \Rightarrow_{(x^1_{i(0)},Y^1_{i(0)+1})} X''_{q+1} \cdots \Rightarrow_{(x^1_{i(e-1)},Y^1_{i(e-1)+1})} X''_{q+e} \\
& \qquad \vdots \\
& \Rightarrow_{(x^2_{i(0)},Y^2_{i(0)+1})} X''_{q+e+1} \cdots \Rightarrow_{(x^2_{i(e-1)},Y^2_{i(e-1)+1})} X''_{q+2e} \\
& \qquad \vdots \\
& \Rightarrow_{(x^m_{i(0)},Y^m_{i(0)+1})} X''_{q+(m-1)e+1} \cdots \Rightarrow_{(x^m_{i(e-1)},Y^m_{i(e-1)+1})} X''_{q+me} \\
& \qquad \vdots \\
& \Rightarrow_{(x^m_q,Y_{q+1})} X''_{q+me+1} \cdots \Rightarrow_{(x_{q+1},Y_{q+2})} X''_{q+me+2} \\
& \cdots \Rightarrow_{(x_{n-1},Y_n)} X''_{n+me}
\end{aligned}
$$

(*)

We call the derivation (*) *m times pumped derivation* and denote $pump(D, x_p, x_q, m)$. Now We can state the pumping lemma for BNLC languages.

Lemma 3.1 (A pumping lemma for BNLC languages) Let D be a rough derivation on (x_p, x_q) and $pump(D, x_p, x_q, m)$ be the m times pumped derivation on (x_p, x_q), where m is an arbitrary positive integer. If the result of D is in G_Δ then the result of $pump(D, x_p, x_q, m)$ is in G_Δ.

4 Properties of L_k

We treat Chomsky normal form problem for u-BNLC languages, i.e., " Whether there is a positive integer k_0, such that for every u-BNLC language L there is a BNLC grammar G with $maxr(G) \leq k_0$ and $L = und(L(G))$?".

We will solve the above problem by proving the following Theorem 4.1:

Theorem 4.1 For each positive integer k, there is a u-BNLC language L_k such that the following condition hold: For all BNLC grammar G with $maxr(G) \leq k$, $und(L(G)) \neq L_k$ hold.

It is not difficult to see that if theorem 4.1 is true then there is no positive integer of the problem. In this section we will construct L_k of theorem 4.1, and we will deal properties on L_k.

For each positive integer k, L_k of theorem 4.1 is construct by the following method.

A construction method of L_k : For each positive integer k, we consider a BNLC grammar $G_k = (\Sigma_k, \Delta_k, P_k, conn_k, Z_{axk})$, where $\Sigma_k = \{a_1, a_2, \cdots, a_k, s\}$, $\Delta_k = \{a_1, a_2, \cdots, a_k\}$, $conn_k(a_i) = a_i$ for all $1 \leq i \leq k$, $conn_k(s) = \Delta_k$, Z_{axk} is a single node with label s, $P_k = \{(s, Y_{1k}), (s, Y_{2k})\}$, where Y_{1k} is complete graph with set of nodes $\{u_1, u_2, \cdots, u_k, u_{k+1}\}$, where $\varphi_{Y_{1k}}(u_i) = a_i$ for all $1 \leq i \leq k$, $\varphi_{Y_{1k}}(u_{k+1}) = s$, Y_{2k} is complete graph with set of nodes $\{v_1, v_2, \cdots, v_k\}$, where $\varphi_{Y_{2k}}(v_i) = a_i$ for all $1 \leq i \leq k$. We difine an unlabeled graph language L_k by $L_k = und(L(G_k))$.

An underlying unlabeled L_k has the following characteristic properties.

Proposition 4.2 Let $k \geq 2$ be an arbitrary integer, e be a positive integer and let H be a graph in $L(G_k)$ such that $\sharp V_H = k \cdot e$. Then

(1) The graph H has exactly e disjoint complete subgraphs with k nodes.

(2) The graph H has exactly k disjoint complete subgraphs with e nodes.

We call the complete subgraphs of (1) in proposition 4.2 *different label group*, and the complete subgraphs of (2) in proposition 4.2 *same label group*.

Same label groups and different label groups has characteristic properties. We will show properties of same label groups and different label groups.

We consider an underlying unlabeled graph H_u in L_k. Let H be a graph such that $und(H) = H_u$, and D be a derivation of H in G_k:

$$D : X_0 \Rightarrow_{(x_0,Y_1)} X_1 \Rightarrow_{(x_1,Y_2)} \cdots \Rightarrow_{(x_{n-1},Y_n)} X_n$$

Let x and y be nodes in H_u. We say that *nodes x and y come under an identicaly same label group in H* if $\varphi_H(x) = \varphi_H(y)$, and that *nodes x and y come under an identicaly different label group in H* if $x \in V_{Y_i}$ and $y \in V_{Y_i}$ for some $1 \leq i \leq n-1$.

Proposition 4.3 Let H_u be an underlying unlabeled graph with $k \cdot e$ nodes in L_k, $F_1, F_2, \cdots, F_e$ be different label groups in H_u, and let $E_1, E_2, \cdots, E_k$ be same label groups in H_u. For any nodes x and y in V_{H_u}, if x and y are adjacent then

(1) Nodes x and y come under an identicaly same label group and do not come under an identicaly different label group. or

(2) Nodes x and y come under an identicaly different label group and do not come under an identicaly same label group.

As a consequence of this proposition, we obtain the following Lemma 4.4.

Lemma 4.4 Let H_u be an underlying unlabeled graph with more than $k^2 (k \geq 2)$ nodes in L_k, and let F be a complete subgraph with more than k nodes in H_u. Then for all nodes x and y in V_F, x and y come under an identicaly same label group.

5 Proof of the theorem

In this section we will show that there is no BNLC grammar G such that $maxr(G) \leq k$ and $L_k = und(L(G))$ by leading contradition.

We assume that there exists a BNLC grammar G such that $maxr(G) \leq k$ and $L_k = und(L(G))$. And we consider a graph $H^1 \in L(G)$ using pumping lemma for a graph $H \in L(G)$. Then we show that $und(H^1) \notin L_k$.

From now on assume that there exists BNLC grammar $G = (\Sigma, \Delta, P, conn, Z_{ax})$ such that $L_k = und(L(G))$ and $maxr(G) \leq k$. Let f be $\sharp\Sigma$, g be $\sharp\Delta$, H be a graph in $L(G)$ with $k(k+2+f-g)(k-1)$ nodes, and let D be a derivation of H in G. By the supposition, $und(H) \in L_k$. Hence by the (2) of proposition 4.2 $und(H)$ has k complete subgraphs with $(k+2+f-g)(k-1)$ nodes. We denote the k complete subgraphs in the graph H by $E_1, E_2, \cdots, E_k$. (For all $1 \leq j \leq k$ complete subgraph in $und(H)$ that correspond with complete subgraph E_j in H is also denoted by E_j.) In the derivation D we denote the nonterminal node that yield jth created nodes in E_i by y^i_j. For convenience we also denote the nonterminal node that yield last created nodes in E_i by $y^i_{e(i)}$. Then for each $1 \leq i \leq k$, set of nonteminal nodes $\{y^i_1, y^i_2, \cdots, y^i_{e(i)}\}$ is denoted by N_i.

Remarks.

(1) If a nonterminal node y yield terminal nodes $u \in E_i$ and $v \in E_j$, then $y \in N_i \cap N_j$.

(2) For $1 \leq i \leq k$, $e(i) \geq k + f - g + 2$.

For each $1 \leq i \leq k$, we denote by N'_i set $\{y^i_n \mid y^i_n \in N_i, k+1 < n \leq e(i)\}$. If a graph X in the derivation D has a node $x \in N'_i$ then X has at least $k+1$ nodes of E_i. Because at least $k+1$ nonterminal nodes in N'_i must be rewriten to rewrite nonterminal node y^i_{k+2}. For $1 \leq i \leq k$ by $\sharp N'_i \geq f - g + 1$ there exists at least a pair of nodes with identicaly nonterminal label, i.e., there exist a pair of nodes $y^i_{h(i)}, y^i_{t(i)}$ such that $\varphi_D(y^i_{h(i)}) = \varphi_D(y^i_{t(i)}) \in \Sigma - \Delta$, where $k+1 < h(i) < t(i) \leq e(i)$. Then for all positive integer m and each $1 \leq i \leq k$ we consider a derivation $pump(D, y^i_{h(i)}, y^i_{t(i)}, m)$ that is constructed by $D_{i\ copy1}, D_{i\ copy2}, \cdots, D_{i\ copym} \in [orig(D, y^i_{h(i)}, y^i_{t(i)})]$. Let $h_{i\ j}$ is isomorophism from $V_{D_{i\ copyj}}$ to $V_{D_{(y^i_{h(i)}, y^i_{t(i)})}}$ for each $1 \leq j \leq m$.

We denote by $E_i(orig(D, y^i_{h(i)}, y^i_{t(i)}))$ set of nodes $\{y \in V_{orig(D, y^i_{h(i)}, y^i_{t(i)})} \mid y \in E_i\}$, and by $E_i(D_{i\ copyj})$ set of nodes $\{y \in V_{D_{i\ copyj}} \mid h_{i\ j}(y) \in E_i(orig(D, y^i_{h(i)}, y^i_{t(i)}))\}$. For convenience we denote by $x_{[j,i\ h(i)]}, x_{[j,i\ t(i)]}$ respectively node u, v such that $h_{i\ j}(u) = y^i_{h(i)}, h_{i\ j}(v) = y^i_{t(i)}$. For the above $y^i_{h(i)}, y^i_{t(i)}$ $(1 \leq i \leq k)$, the following lemma is satisfied.

Lemma 5.1 There exists sequence $hist_D(y^i_{h(i)}, y^i_{t(i)})$ that hold the following condition:

Condition : For node $y \in hist_D(y^i_{h(i)}, y^i_{t(i)})$, If in the derivation D a graph X has y, then y and all terminal nodes of graph X are adjacent.

We construct graph H^1 as following: Let $hist_D(x_h, x_t)$ be sequence that be obtained by Algorithm A of Lemma 5.1, and $pump(D, x_h, x_t, 1)$ be a derivation that be constructed by the derivation D and $D_{copy1} \in [iter(D, x_h, x_t)]$, and let H^1 be a result of the derivation $pump(D, x_h, x_t, 1)$. Then the following lemma hold.

Lemma 5.2 $und(H^1) \notin L_k$.

References

Ehrenfeucht, A., Main, M., and Rozenberg, G. (1984), Restrictions on NLC grammars, *Theoret. Comput. Sci.* **31**, 211-223.

Harary, F. (1969), "Graph Theory,"Addison-Wesley, Reading, Mass.

Hopcroft, J. E. and Ullman, J. D. (1979), "Introduction to Automata Theory, Languages and Computation,"Addison-Wesley, Reading.

Janssens, D., and Rozenberg, G. (1980 a), On the structure of node label controlled graph languages, *Inform. Sci.* **20**, 191-216.

Janssens, D., and Rozenberg, G. (1980 b), Restrictions, extenstions and variations of NLC grammars, *Inform. Sci.* **20**, 217-244.

Janssens, D., and Rozenberg, G., and Welzl, E. (1984), The bounded degree problem for NLC grammars is decidable, *J. Comput. System. Sci.*, **35**, 415-422.

Rozenberg, G., and Welzl, E. (1986 a), Boundary NLC graph grammars - basic definitions, normal forms, and complexity, *Inform. and Control* **69**, 136-167.

Rozenberg, G., AND Welzl, E. (1986 b), Graph theoretic closure properties of boundary NLC graph languages, *Acta inform.*, **23**, 289-309.